

Local instance MySQL80 ×

File Edit View Query Database Server Tools Scripting Help



Navigator

SCHEMAS

Filter objects

customer_reviews_project

- Tables
- Views
- Stored Procedures
- Functions

employees

sys

temp

Administration Schemas

Information

No object selected

Object Info

Session

main

SQL File 4*

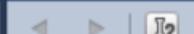
SQL File 4* ×



Limit to 1000 rows

```
1 -- Creating Database
2 • CREATE DATABASE customer_reviews_project;
3 • use customer_reviews_project;
```

SQLAdditions



Jump to

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

Context Help Snippets

Output

Action Output

#	Time	Action
29	11:15:14	CREATE DATABASE customer_reviews_project
30	11:15:14	use customer_reviews_project

Message

1 row(s) affected

0 row(s) affected

Duration / Fetch

0.000 sec

0.000 sec

MySQL Workbench

Local instance MySQL80 ×

File Edit View Query Database Server Tools Scripting Help

Schemas

customer_reviews_project

- Tables
- Views
- Stored Procedures
- Functions

employees

sys

temp

Navigator

SQL File 4*

SQL File 4*

main

Limit to 1000 rows

1 -- Creating Database
2 • CREATE DATABASE customer_reviews_project;
3 • use customer_reviews_project;
4
5 -- Inserting table from CSV file
6

Table Data Import

Select File to Import

Table Data Import allows you to easily import CSV, JSON datafiles.
You can also create destination table on the fly.

File Path: E:\Documents\customer_reviews.csv

Browse...

Administration Schemas

Information

Schema:
customer_reviews_project

Object Info Session

SQLAdditions

Jump to

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

Context Help Snippets

Duration / Fetch
0.000 sec
0.000 sec

Navigator

main SQL File 4* SQL File 4* x SQLAdditions

SCHEMAS

customer_reviews_project

Tables

customer_reviews

Columns

- Date
- ID
- Name
- Region
- Rating
- Product
- Quantity
- Price Per Unit

Indexes

Foreign Keys

Triggers

Views

Stored Procedures

Functions

employees

sys

temp

Administration Schemas

Information

Schema: customer_reviews_project

Result Grid | Filter Rows: Export: Wrap Cell Content: Result Grid

1 -- Creating Database
2 • CREATE DATABASE customer_reviews_project;
3 • use customer_reviews_project;
4
5 -- Inserting table from CSV file
6
7 -- Checking imported table and its data
8 • SELECT * FROM customer_reviews;

	Date	ID	Name	Region	Rating	Product	Quantity	Price Per Unit
▶	31-01-2021 00:00	1	John Smith	North	Good	Magic Wand	10	\$20.00
	28-02-2021 00:00	2	Jane Doe	East	Excelent	Unicorn Horn	15	\$10.00
	31-03-2021 00:00	3	Mike Tyson	West	Poor	Boxing Gloves	0	inf
	30-04-2021 00:00	4	Anna Belle	South	Average	Fairy Dust	25	\$10.00
	31-05-2021 00:00	5	Chris P. Bacon	East	Good	Bacon Scented Candle	30	\$16.67
	30-06-2021 00:00	6	Peter Parker		Excelent	Web Shooter	0	inf
	31-07-2021 00:00	7	Mary Jane	West	Poor	Potent Potion	35	\$10.00
	31-08-2021 00:00	8	Bruce Wayne	South	Average	Bat Signal	40	\$15.00
	30-09-2021 00:00	9	Clark Kent	East	Good	Glasses with X-ray Vision	45	\$12.22
	31-10-2021 00:00	10	Diana Prince	North	Excelent	Lasso of Truth	50	\$14.00
	30-11-2021 00:00	11	Tony Stark	West	Poor	Iron Man Suit	5	\$160.00
	31-12-2021 00:00	12	Steve Rogers	South	Average	Captain America Shield	20	\$45.00
	31-01-2022 00:00	13	Natasha Rom...	East	Good	Black Widow's Bite	0	inf
	28-02-2022 00:00	14	Bruce Banner		Excelent	Gamma Radiation Serum	30	\$36.67
	30-04-2021 00:00	4	Anna Belle	South	Average	Fairy Dust	25	\$10.00
	31-05-2021 00:00	5	Chris P. Bacon	East	Good	Bacon Scented Candle	30	\$16.67

customer_reviews 7 x Read Only Context Help Snippets

Output

Action Output

#	Time	Action	Message	Duration / Fetch
36	11:16:47	DEALLOCATE PREPARE stmt	OK	0.000 sec
37	11:17:19	SELECT * FROM customer_reviews LIMIT 0, 1000	31 row(s) returned	0.000 sec / 0.000 sec

Object Info Session

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

```
9  
10    -- Data Cleaning  
11  
12    -- Finding Duplicates  
13 • with cte_for_checking_duplicates as (  
14        select *, row_number() over(PARTITION BY Date, ID, Name, Region, Rating, Product, Quantity, `Price Per Unit`) as `Row Number`  
15        from customer_reviews  
16    )  
17    select * from cte_for_checking_duplicates  
18    where `Row Number` > 1;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

	Date	ID	Name	Region	Rating	Product	Quantity	Price Per Unit	Row Number
▶	30-04-2021 00:00	4	Anna Belle	South	Average	Fairy Dust	25	\$10.00	2
	30-06-2021 00:00	6	Peter Parker		Excellent	Web Shooter	0	inf	2
	31-05-2021 00:00	5	Chris P. Bacon	East	Good	Bacon Scented Candle	30	\$16.67	2

Result 2 × Read Only

```
19
20    -- Deleting Duplicates
21
22    /*
23        To delete duplicates we have to create new table with all values from 'customer_reviews' including 'Row Number' column
24        So we will create new table named as customer_reviews_2 with all columns from 'customer_reviews' including 'Row Number' column.
25        */
26
27 • CREATE TABLE `customer_reviews_2` (
28     `Date` text,
29     `ID` int DEFAULT NULL,
30     `Name` text,
31     `Region` text,
32     `Rating` text,
33     `Product` text,
34     `Quantity` int DEFAULT NULL,
35     `Price Per Unit` text,
36     `Row Number` int
37 ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
38 •     SELECT * from customer_reviews_2;
```

< >

Result Grid | Filter Rows: Export: Wrap Cell Content:

Date	ID	Name	Region	Rating	Product	Quantity	Price Per Unit	Row Number
------	----	------	--------	--------	---------	----------	----------------	------------

```
39
40 -- Inserting values in 'customer_reviews_2' table from 'customer_reviews' table with values for 'Row Number' column
41
42 • INSERT into customer_reviews_2
43 select *, row_number() over(PARTITION BY Date, ID, Name, Region, Rating, Product, Quantity, `Price Per Unit`) as `Row Number`
44     from customer_reviews;
45
46 • SELECT * from customer_reviews_2;
47
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

	Date	ID	Name	Region	Rating	Product	Quantity	Price Per Unit	Row Number
▶	28-02-2021 00:00	2	Jane Doe	East	Excelent	Unicorn Horn	15	\$10.00	1
	28-02-2022 00:00	14	Bruce Banner		Excelent	Gamma Radiation Serum	30	\$36.67	1
	28-02-2023 00:00	26	Thor Odinson	Asgard	Worthy	Mjolnir	80	\$28.75	1
	30-04-2021 00:00	4	Anna Belle	South	Average	Fairy Dust	25	\$10.00	1
	30-04-2021 00:00	4	Anna Belle	South	Average	Fairy Dust	25	\$10.00	2
	30-04-2022 00:00	16	Phil Coulson		Average	Agent ID Card	0	inf	1

customer_reviews_2 4 ×

Result Grid | Form Editor

Read Only

```
47  
48 -- Checking duplicates again in new table 'customer_reviews_2'  
49 • select * from customer_reviews_2  
50 where `Row Number` > 1;  
51
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

	Date	ID	Name	Region	Rating	Product	Quantity	Price Per Unit	Row Number
▶	30-04-2021 00:00	4	Anna Belle	South	Average	Fairy Dust	25	\$10.00	2
	30-06-2021 00:00	6	Peter Parker		Excellent	Web Shooter	0	inf	2
	31-05-2021 00:00	5	Chris P. Bacon	East	Good	Bacon Scented Candle	30	\$16.67	2



Result Grid



Form Editor



```
51  
52     -- Total records before deleting  
53 • select count(*) as `Total Records` from customer_reviews_2;  
54
```

< >

Result Grid | Filter Rows: Export: Wrap Cell Content:

Total Records
31

Result 9 × Read Only

Result Grid

Form Editor

```
54  
55      -- Now deleting duplicates  
56 • DELETE from customer_reviews_2  
57      where `Row Number` > 1;
```

< > ▾

Output



Action Output

#	Time	Action	Message
✓	16 11:48:12	select count(*) as 'Total Records' from customer_reviews_2 LIMIT 0, 1000	1 row(s) returned
✓	17 11:49:22	DELETE from customer_reviews_2 where 'Row Number' > 1	3 row(s) affected

```
58  
59      -- Total records after deleting of duplicates  
60 •  select count(*) as `Total Records` from customer_reviews_2;  
61
```

Result Grid | Filter Rows: [] | Export: [] | Wrap Cell Content: []

Total Records
28

Result 10 ×

! Read Only

The screenshot shows a database query results interface. At the top, there is a code editor window containing SQL code. Below it is a toolbar with buttons for 'Result Grid', 'Filter Rows', 'Export', and 'Wrap Cell Content'. The main area displays a table with one row. The table has a single column labeled 'Total Records' with the value '28'. On the right side, there is a vertical sidebar with icons for 'Result Grid' (selected), 'Form Editor', and a downward arrow. At the bottom, there is a status bar with the text 'Result 10 ×' and a 'Read Only' indicator.

```
61  
62      -- Checking and Correcting important misspelled words  
63 •  SELECT * from customer_reviews_2;  
64 •  SELECT distinct(Rating) from customer_reviews_2;  
65
```

Result Grid | Filter Rows: Export: Wrap Cell Content:

Rating
Excellent
Worthy
Average
Leader
Good
Poor
Mischief
Spy

Result 19 Read Only

Result Grid | Form Editor | Field Types | Query Stats

```
65  
66 • update customer_reviews_2  
67     set Rating = 'Excellent'  
68     where Rating = 'Excelent';  
69  
70 • SELECT distinct(Rating) from customer_reviews_2;
```

< [] Filter Rows: [] Export: [] Wrap Cell Co

Rating
Excellent
Worthy
Average
Leader
Good
Poor
Mischief
Spy

```
71
72      -- Checking and Removing space which present before and after name
73 •  select Name, trim(Name)
74  from customer_reviews_2;
75
76      -- see this name has space after name. 'John Smith   ' so after trim it will become 'John Smith'
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

Name	trim(Name)
Jane Doe	Jane Doe
Bruce Banner	Bruce Banner
Thor Odinson	Thor Odinson
Anna Belle	Anna Belle
Phil Coulson	Phil Coulson
Steve Rogers	Steve Rogers
Peter Parker	Peter Parker
Howard Stark	Howard Stark
Clark Kent	Clark Kent
Kurt Busiek	Kurt Busiek
Tony Stark	Tony Stark
Roger Stern	Roger Stern
▶ John Smith	John Smith
Natasha Ro...	Natasha Ro...
Loki Laufeyson	Loki Laufeyson
Mike Tyson	Mike Tyson
Nick Fury	Nick Fury

Result 38 × Read Only

```
71  
72      -- Checking and Removing space which present before and after name  
73 •  select Name, trim(Name)  
74  from customer_reviews_2;  
75  
76      -- see this name has space after name. 'John Smith   ' so after trim it will become 'John Smith'  
77      -- now updating values using trim  
78 •  Update customer_reviews_2  
79      set Name = trim(Name);
```

Output

Action Output

#	Time	Action	Message
49	12:24:09	select Name, trim(Name) from customer_reviews_2 LIMIT 0, 1000	28 row(s) returned
50	12:27:05	Update customer_reviews_2 set Name = trim(Name)	1 row(s) affected Rows matched: 28 Changed: 1 Warnings: 0

```
80
81      -- some name has space between first name and last name
82 •  select Name from customer_reviews_2;
83      -- like this name has space more than one between first name and last name 'Peter Parker'
84
```

Result Grid | Filter Rows: [] Export: [] Wrap Cell Content: []

Name
Jane Doe
Bruce Banner
Thor Odinson
Anna Belle
Phil Coulson
Steve Rogers
Peter Parker
Howard Stark
Clark Kent
Kurt Busiek
Tony Stark
Roger Stern
John Smith
Natasha Ro...
Loki Laufeyson

customer_reviews_2 39 ×

Result Grid | Form Editor | Field Types | Query Stats

Read Only

```
80
81      -- some name has space between first name and last name
82 •  select Name from customer_reviews_2;
83 /*
84 like this name has space more than one betwween first name and last name 'Peter Parker'.
85 to remove space more than one between first name and last name we have to use following query multiple times until we get all
86 spaces removed .
87 */
88 •  update customer_reviews_2
89     set Name = replace(Name, ' ', ' ')
90     where Name like '% %';
91 -- now here we can see 3 rows affected and 3 changed, we will run this query again and again till it shows 0 rows affected.
```



Output

Action Output

#	Time	Action	Message
51	12:29:07	select Name from customer_reviews_2 LIMIT 0, 1000	28 row(s) returned
52	12:34:10	update customer_reviews_2 set Name = replace(Name, ' ', '') where Name like '% %'	3 row(s) affected Rows matched: 3 Changed: 3 Warnings: 0

```
83      /*
84      like this name has space more than one between first name and last name 'Peter Parker'.
85      to remove space more than one between first name and last name we have to use following query multiple times until we get all
86      spaces removed .
87      */
88 • update customer_reviews_2
89 set Name = replace(Name, ' ', ' ')
90 where Name like '% %';
91 -- now here we can see 3 rows affected and 3 changed, we will run this query again and again till it shows 0 rows affected.
```

Output

Action Output

#	Time	Action	Message
52	12:34:10	update customer_reviews_2 set Name = replace(Name, ' ', ' ') where Name like '% %'	3 row(s) affected Rows matched: 3 Changed: 3 Warnings: 0
53	12:36:50	update customer_reviews_2 set Name = replace(Name, ' ', ' ') where Name like '% %'	3 row(s) affected Rows matched: 3 Changed: 3 Warnings: 0

```
83 *
84 like this name has space more than one between first name and last name 'Peter Parker'.
85 to remove space more than one between first name and last name we have to use following query multiple times until we get all
86 spaces removed .
87 */
88 • update customer_reviews_2
89 set Name = replace(Name, ' ', ' ')
90 where Name like '% %';
91 -- now here we can see 3 rows affected and 3 changed, we will run this query again and again till it shows 0 rows affected.
92 -- now its done. 0 rows affected means no more name present which has more than one space between it.
```

Output

Action Output

#	Time	Action	Message
53	12:36:50	update customer_reviews_2 set Name = replace(Name, ' ', ' ') where Name like '% %'	3 row(s) affected Rows matched: 3 Changed: 3 Warnings: 0
54	12:37:23	update customer_reviews_2 set Name = replace(Name, ' ', ' ') where Name like '% %'	0 row(s) affected Rows matched: 0 Changed: 0 Warnings: 0

```
88 • update customer_reviews_2
89   set Name = replace(Name, ' ', '')
90   where Name like '% %';
91   -- now here we can see 3 rows affected and 3 changed, we will run this query again and again till it shows 0 rows affected.
92   -- now its done. 0 rows affected means no more name present which has more than one space between it.
93 • select Name from customer_reviews_2;
94   -- we can see result
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

Name
Jane Doe
Bruce Banner
Thor Odinson
Anna Belle
Phil Coulson
Steve Rogers
Peter Parker
Howard Stark
Clark Kent
Kurt Busiek
Tony Stark
Roger Stern
John Smith
Natasha Romanoff
Loki Laufey

customer_review

Output: Action Output

#	Time
54	12:37:
55	12:39:

Copy Field (unquoted)

Paste Row

Capitalize Text

lowercase Text

UPPERCASE Text

Message

Message
update customer_reviews_2 set Name = replace(Name, ' ', '') where Name like '% %'
0 row(s) affected Rows matched: 0 Changed: 0 Warnings: 0
000
28 row(s) returned

Result Grid | Form Editor | Field Types | Query Stats | Read Only

```
88 • update customer_reviews_2
89 set Name = replace(Name, ' ', '')
90 where Name like '% %';
91 -- now here we can see 3 rows affected and 3 changed, we will run this query again and again till it shows 0 rows affected.
92 -- now its done. 0 rows affected means no more name present which has more than one space between it.
93 • select Name from customer_reviews_2;
94 -- we can see result 'Peter Parker'
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

Name
Jane Doe
Bruce Banner
Thor Odinson
Anna Belle
Phil Coulson
Steve Rogers
Peter Parker
Howard Stark
Clark Kent
Kurt Busiek
Tony Stark
Roger Stern
John Smith
Natasha Romanoff
Loki Laufeyson

customer_reviews_2 40 x Read Only

Output

Action Output

#	Time	Action	Message
54	12:37:23	update customer_reviews_2 set Name = replace(Name, ' ', '') where Name like '% %'	0 row(s) affected Rows matched: 0 Changed: 0 Warnings: 0
55	12:39:21	select Name from customer_reviews_2 LIMIT 0, 1000	28 row(s) returned

```
95
96    -- Cheking regin column for accuracy.
97 • select distinct(Region) from customer_reviews_2;
98    -- so here are some empty values and incorrect values like 'Asgard'
```

Result Grid | Filter Rows: Export: Wrap Cell Content:

Region
East
Asgard
South
North
West



Result
Grid



Form
Editor



Field
Types



Query
Stats

```
95
96    -- Cheking region column for accuracy.
97 • select distinct(Region) from customer_reviews_2;
98     -- so here are some empty values and incorrect values like 'Asgard'
99     -- so we will replace those with 'NA' - Not Available
100 • UPDATE customer_reviews_2
101    set Region = 'NA'
102   where Region = 'Asgard' or Region = '';
103
104 • select distinct(Region) from customer_reviews_2;
```

< >

Result Grid | Filter Rows: Export: Wrap Cell Content:

Region
East
NA
South
North
West

Result Grid

Form Editor

Field Types

Query Stats

Read Only

```
106 -- replacing 'inf' value to 0 from 'Price Per Unit' column
```

```
107 • SELECT * from customer_reviews_2;
```

Result Grid | Filter Rows: Export: Wrap Cell Content:

	Date	ID	Name	Region	Rating	Product	Quantity	Price Per Unit	Row Number
▶	28-02-2021 00:00	2	Jane Doe	East	Excellent	Unicorn Horn	15	\$10.00	1
	28-02-2022 00:00	14	Bruce Banner	NA	Excellent	Gamma Radiation Serum	30	\$36.67	1
	28-02-2023 00:00	26	Thor Odinson	NA	Worthy	Mjolnir	80	\$28.75	1
	30-04-2021 00:00	4	Anna Belle	South	Average	Fairy Dust	25	\$10.00	1
	30-04-2022 00:00	16	Phil Coulson	NA	Average	Agent ID Card	0	inf	1
	30-04-2023 00:00	28	Steve Rogers	South	Leader	Leadership Manual	85	\$29.41	1
	30-06-2021 00:00	6	Peter Parker	NA	Excellent	Web Shooter	0	inf	1
	30-06-2022 00:00	18	Howard Stark	North	Excellent	Arc Reactor	45	\$33.33	1
	30-09-2021 00:00	9	Clark Kent	East	Good	Glasses with X-ray Vision	45	\$12.22	1
	30-09-2022 00:00	21	Kurt Busiek	East	Good	Comic Book	60	\$30.00	1
	30-11-2021 00:00	11	Tony Stark	West	Poor	Iron Man Suit	5	\$160.00	1
	30-11-2022 00:00	23	Roger Stern	West	Poor	Notepads	65	\$30.77	1

Result Grid | Form Editor | Field Types

```
106 -- replacing 'inf' value to 0 from 'Price Per Unit' column
107 • SELECT * from customer_reviews_2;
108
109 • UPDATE customer_reviews_2
110   set `Price Per Unit` = 0
111   where `Price Per Unit` = 'inf';
```

Result Grid | Filter Rows: Export: Wrap Cell Content:

	Date	ID	Name	Region	Rating	Product	Quantity	Price Per Unit	Row Number
▶	28-02-2021 00:00	2	Jane Doe	East	Excellent	Unicorn Horn	15	\$10.00	1
	28-02-2022 00:00	14	Bruce Banner	NA	Excellent	Gamma Radiation Serum	30	\$36.67	1
	28-02-2023 00:00	26	Thor Odinson	NA	Worthy	Mjolnir	80	\$28.75	1
	30-04-2021 00:00	4	Anna Belle	South	Average	Fairy Dust	25	\$10.00	1
	30-04-2022 00:00	16	Phil Coulson	NA	Average	Agent ID Card	0	0	1
	30-04-2023 00:00	28	Steve Rogers	South	Leader	Leadership Manual	85	\$29.41	1
	30-06-2021 00:00	6	Peter Parker	NA	Excellent	Web Shooter	0	0	1
	30-06-2022 00:00	18	Howard Stark	North	Excellent	Arc Reactor	45	\$33.33	1
	30-09-2021 00:00	9	Clark Kent	East	Good	Glasses with X-ray Vision	45	\$12.22	1
	30-09-2022 00:00	21	Kurt Busiek	East	Good	Comic Book	60	\$30.00	1
	30-11-2021 00:00	11	Tony Stark	West	Poor	Iron Man Suit	5	\$160.00	1
	30-11-2022 00:00	23	Roger Stern	West	Poor	Notepads	65	\$30.77	1

Result Grid | Form Editor | Field Types

```
112
113    -- Correcting data types of columns
114 • DESCRIBE customer_reviews_2;
```

< >

Result Grid | Filter Rows: [] | Export: [] | Wrap Cell Content: []

	Field	Type	Null	Key	Default	Extra
▶	Date	text	YES		NULL	
	ID	int	YES		NULL	
	Name	text	YES		NULL	
	Region	text	YES		NULL	
	Rating	text	YES		NULL	
	Product	text	YES		NULL	
	Quantity	int	YES		NULL	
	Price Per Unit	text	YES		NULL	
	Row Number	int	YES		NULL	

Result Grid | Form Editor | Field Types

```
116 -- we can see 'Date' column has 'text' data type which is incorrect.  
117 • SELECT * from customer_reviews_2;  
118  
119 -- we have to remove 00:00 time.  
120 • update customer_reviews_2  
121 set Date = replace(Date, '00:00', '')  
122 where Date like '%00:00%';  
123  
124 • SELECT * from customer_reviews_2;
```

< []

Result Grid | Filter Rows: [] | Export: [] | Wrap Cell Content: []

	Date	ID	Name	Region	Rating	Product	Quantity	Price Per Unit	Row Number
▶	28-02-2021	2	Jane Doe	East	Excellent	Unicorn Horn	15	\$10.00	1
	28-02-2022	14	Bruce Banner	NA	Excellent	Gamma Radiation Serum	30	\$36.67	1
	28-02-2023	26	Thor Odinson	NA	Worthy	Mjolnir	80	\$28.75	1
	30-04-2021	4	Anna Belle	South	Average	Fairy Dust	25	\$10.00	1
	30-04-2022	16	Phil Coulson	NA	Average	Agent ID Card	0	0	1
	30-04-2023	28	Steve Rogers	South	Leader	Leadership Manual	85	\$29.41	1
	30-06-2021	6	Peter Parker	NA	Excellent	Web Shooter	0	0	1
	30-06-2022	18	Howard Stark	North	Excellent	Arc Reactor	45	\$33.33	1
	30-09-2021	9	Clark Kent	East	Good	Glasses with X-ray Vision	45	\$12.22	1
	30-09-2022	21	Kurt Busiek	East	Good	Comic Book	60	\$30.00	1
	30-11-2021	11	Tony Stark	West	Poor	Iron Man Suit	5	\$160.00	1
	30-11-2022	23	Roger Stern	West	Poor	Notebooks	65	\$30.77	1

```
125
126    -- trimming any space from date column
127 • update customer_reviews_2
128     set Date = trim(Date);
129
130 • SELECT * from customer_reviews_2;
```

Result Grid									
	Date	ID	Name	Region	Rating	Product	Quantity	Price Per Unit	Row Number
▶	28-02-2021	2	Jane Doe	East	Excellent	Unicorn Horn	15	\$10.00	1
	28-02-2022	14	Bruce Banner	NA	Excellent	Gamma Radiation Serum	30	\$36.67	1
	28-02-2023	26	Thor Odinson	NA	Worthy	Mjolnir	80	\$28.75	1
	30-04-2021	4	Anna Belle	South	Average	Fairy Dust	25	\$10.00	1
	30-04-2022	16	Phil Coulson	NA	Average	Agent ID Card	0	0	1
	30-04-2023	28	Steve Rogers	South	Leader	Leadership Manual	85	\$29.41	1
	30-06-2021	6	Peter Parker	NA	Excellent	Web Shooter	0	0	1
	30-06-2022	18	Howard Stark	North	Excellent	Arc Reactor	45	\$33.33	1
	30-09-2021	9	Clark Kent	East	Good	Glasses with X-ray Vision	45	\$12.22	1
	30-09-2022	21	Kurt Busiek	East	Good	Comic Book	60	\$30.00	1
	30-11-2021	11	Tony Stark	West	Poor	Iron Man Suit	5	\$160.00	1
	30-11-2022	23	Roger Stern	West	Poor	Notenads	65	\$30.77	1

```
131  
132      -- Now correcting inserted date format to change data type of Date column  
133      -- here we can see the correct format for date.  
134 •  select Date, str_to_date(Date, '%d-%m-%y') as `Correct Date format` from customer_reviews_2;
```

Result Grid | Filter Rows: Export: Wrap Cell Content:

	Date	Correct Date format
▶	28-02-2021	2020-02-28
	28-02-2022	2020-02-28
	28-02-2023	2020-02-28
	30-04-2021	2020-04-30
	30-04-2022	2020-04-30
	30-04-2023	2020-04-30
	30-06-2021	2020-06-30
	30-06-2022	2020-06-30
	30-09-2021	2020-09-30
	30-09-2022	2020-09-30
	30-11-2021	2020-11-30
	30-11-2022	2020-11-30

Result Grid | Form Editor | Field Types

```
135
136    -- correcting date format
137
138 • update customer_reviews_2
139     set `Date` = str_to_date(`Date`, '%d-%m-%Y')
140     WHERE str_to_date(`Date`, '%d-%m-%Y') is not null;
141
142 • SELECT * from customer_reviews_2;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

	Date	ID	Name	Region	Rating	Product	Quantity	Price Per Unit	Row Number
▶	2021-02-28	2	Jane Doe	East	Excellent	Unicorn Horn	15	\$10.00	1
	2022-02-28	14	Bruce Banner	NA	Excellent	Gamma Radiation Serum	30	\$36.67	1
	2023-02-28	26	Thor Odinson	NA	Worthy	Mjolnir	80	\$28.75	1
	2021-04-30	4	Anna Belle	South	Average	Fairy Dust	25	\$10.00	1
	2022-04-30	16	Phil Coulson	NA	Average	Agent ID Card	0	0	1
	2023-04-30	28	Steve Rogers	South	Leader	Leadership Manual	85	\$29.41	1
	2021-06-30	6	Peter Parker	NA	Excellent	Web Shooter	0	0	1
	2022-06-30	18	Howard Stark	North	Excellent	Arc Reactor	45	\$33.33	1
	2021-09-30	9	Clark Kent	East	Good	Glasses with X-ray Vision	45	\$12.22	1
	2022-09-30	21	Kurt Busiek	East	Good	Comic Book	60	\$30.00	1
	2021-11-30	11	Tony Stark	West	Poor	Iron Man Suit	5	\$160.00	1
	2022-11-30	23	Roger Stern	West	Poor	Notepads	65	\$30.77	1

Result Grid | Form Editor | Field Types

```
143
144 -- Now changing data type of 'Date' column
145 • alter table customer_reviews_2
146   modify COLUMN Date DATE;
147
148 • DESCRIBE customer_reviews_2;
```

Result Grid | Filter Rows: Export: Wrap Cell Content:

Field	Type	Null	Key	Default	Extra
Date	date	YES		NULL	
ID	int	YES		NULL	
Name	text	YES		NULL	
Region	text	YES		NULL	
Rating	text	YES		NULL	
Product	text	YES		NULL	
Quantity	int	YES		NULL	
Price Per Unit	text	YES		NULL	
Row Number	int	YES		NULL	

Result Grid | Form Editor | Field Types

149

150 -- now to change data type of 'Price Per Unit' to int we have to remove '\$' Sign

151 • SELECT * from customer_reviews_2;

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

	Date	ID	Name	Region	Rating	Product	Quantity	Price Per Unit	Row Number
▶	2021-02-28	2	Jane Doe	East	Excellent	Unicorn Horn	15	\$10.00	1
	2022-02-28	14	Bruce Banner	NA	Excellent	Gamma Radiation Serum	30	\$36.67	1
	2023-02-28	26	Thor Odinson	NA	Worthy	Mjolnir	80	\$28.75	1
	2021-04-30	4	Anna Belle	South	Average	Fairy Dust	25	\$10.00	1
	2022-04-30	16	Phil Coulson	NA	Average	Agent ID Card	0	0	1
	2023-04-30	28	Steve Rogers	South	Leader	Leadership Manual	85	\$29.41	1
	2021-06-30	6	Peter Parker	NA	Excellent	Web Shooter	0	0	1
	2022-06-30	18	Howard Stark	North	Excellent	Arc Reactor	45	\$33.33	1
	2021-09-30	9	Clark Kent	East	Good	Glasses with X-ray Vision	45	\$12.22	1
	2022-09-30	21	Kurt Busiek	East	Good	Comic Book	60	\$30.00	1
	2021-11-30	11	Tony Stark	West	Poor	Iron Man Suit	5	\$160.00	1
	2022-11-30	23	Roger Stern	West	Poor	Notebooks	65	\$30.77	1

Result Grid | Form Editor | Field Types

149

150 -- now to change data type of 'Price Per Unit' to int we have to remove '\$' Sign

151 • update customer_reviews_2

152 set `Price Per Unit` = replace(`Price Per Unit`, '\$', '')

153 where `Price Per Unit` like '%\$%';

154

155 • SELECT * from customer_reviews_2;

Result Grid | Filter Rows: Export: Wrap Cell Content:

	Date	ID	Name	Region	Rating	Product	Quantity	Price Per Unit	Row Number
▶	2021-02-28	2	Jane Doe	East	Excellent	Unicorn Horn	15	10.00	1
	2022-02-28	14	Bruce Banner	NA	Excellent	Gamma Radiation Serum	30	36.67	1
	2023-02-28	26	Thor Odinson	NA	Worthy	Mjolnir	80	28.75	1
	2021-04-30	4	Anna Belle	South	Average	Fairy Dust	25	10.00	1
	2022-04-30	16	Phil Coulson	NA	Average	Agent ID Card	0	0	1
	2023-04-30	28	Steve Rogers	South	Leader	Leadership Manual	85	29.41	1
	2021-06-30	6	Peter Parker	NA	Excellent	Web Shooter	0	0	1
	2022-06-30	18	Howard Stark	North	Excellent	Arc Reactor	45	33.33	1
	2021-09-30	9	Clark Kent	East	Good	Glasses with X-ray Vision	45	12.22	1
	2022-09-30	21	Kurt Busiek	East	Good	Comic Book	60	30.00	1
	2021-11-30	11	Tony Stark	West	Poor	Iron Man Suit	5	160.00	1
	2022-11-30	23	Roger Stern	West	Poor	Notepad	65	30.77	1

```
156
157    -- triming spaces of values from 'Price Per Unit' column.
158 • update customer_reviews_2
159     set `Price Per Unit` = trim(`Price Per Unit`);
160
161 • SELECT * from customer_reviews_2;
```

Result Grid | Filter Rows: Export: Wrap Cell Content:

	Date	ID	Name	Region	Rating	Product	Quantity	Price Per Unit	Row Number
▶	2021-02-28	2	Jane Doe	East	Excellent	Unicorn Horn	15	10.00	1
	2022-02-28	14	Bruce Banner	NA	Excellent	Gamma Radiation Serum	30	36.67	1
	2023-02-28	26	Thor Odinson	NA	Worthy	Mjolnir	80	28.75	1
	2021-04-30	4	Anna Belle	South	Average	Fairy Dust	25	10.00	1
	2022-04-30	16	Phil Coulson	NA	Average	Agent ID Card	0	0	1
	2023-04-30	28	Steve Rogers	South	Leader	Leadership Manual	85	29.41	1
	2021-06-30	6	Peter Parker	NA	Excellent	Web Shooter	0	0	1
	2022-06-30	18	Howard Stark	North	Excellent	Arc Reactor	45	33.33	1
	2021-09-30	9	Clark Kent	East	Good	Glasses with X-ray Vision	45	12.22	1
	2022-09-30	21	Kurt Busiek	East	Good	Comic Book	60	30.00	1
	2021-11-30	11	Tony Stark	West	Poor	Iron Man Suit	5	160.00	1
	2022-11-30	23	Roger Stern	West	Poor	Notenads	65	30.77	1

Result Grid

Form Editor

Field Types

```
162  
163      -- Now changing data type of Column 'Price per Unit' to int  
164 • alter table customer_reviews_2  
165     modify COLUMN `Price Per Unit` int;  
166  
167 • DESCRIBE customer_reviews_2;
```

< >

Field	Type	Null	Key	Default	Extra
Date	date	YES		NULL	
ID	int	YES		NULL	
Name	text	YES		NULL	
Region	text	YES		NULL	
Rating	text	YES		NULL	
Product	text	YES		NULL	
Quantity	int	YES		NULL	
Price Per Unit	int	YES		NULL	
Row Number	int	YES		NULL	

Filter Rows: Export: Wrap Cell Content:



168

```
169      -- declearing 'ID' as a Primary Key and not null  
170 • alter table customer_reviews_2  
171     add Primary key (ID);  
172  
173 • DESCRIBE customer_reviews_2;
```

< []

Result Grid | Filter Rows: [] Export: [] Wrap Cell Content: []

Field	Type	Null	Key	Default	Extra
Date	date	YES		NULL	
ID	int	NO	PRI	NULL	
Name	text	YES		NULL	
Region	text	YES		NULL	
Rating	text	YES		NULL	
Product	text	YES		NULL	
Quantity	int	YES		NULL	
Price Per Unit	int	YES		NULL	
Row Number	int	YES		NULL	

```
174  
175      -- removing 'Row Number Column' because not necessary now  
176 • alter table customer_reviews_2  
177     drop column `Row Number`;  
178  
179 • SELECT * from customer_reviews_2;
```

Result Grid								
	Date	ID	Name	Region	Rating	Product	Quantity	Price Per Unit
▶	2021-01-31	1	John Smith	North	Good	Magic Wand	10	20
	2021-02-28	2	Jane Doe	East	Excellent	Unicorn Horn	15	10
	2021-03-31	3	Mike Tyson	West	Poor	Boxing Gloves	0	0
	2021-04-30	4	Anna Belle	South	Average	Fairy Dust	25	10
	2021-05-31	5	Chris P. Bacon	East	Good	Bacon Scented Candle	30	17
	2021-06-30	6	Peter Parker	NA	Excellent	Web Shooter	0	0
	2021-07-31	7	Mary Jane	West	Poor	Potent Potion	35	10
	2021-08-31	8	Bruce Wayne	South	Average	Bat Signal	40	15
	2021-09-30	9	Clark Kent	East	Good	Glasses with X-ray Vision	45	12
	2021-10-31	10	Diana Prince	North	Excellent	Lasso of Truth	50	14
	2021-11-30	11	Tony Stark	West	Poor	Iron Man Suit	5	160
	2021-12-31	12	Steve Rogers	South	Average	Captain America Shield	20	45

```
180
181    -- Finding some insights
182
183    -- Total Quantity by Region
184 • select Region, sum(Quantity) as `Total Quantity` from customer_reviews_2
185    GROUP BY Region;
```

< >

Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

Region	Total Quantity
North	105
East	190
West	190
South	295
NA	185

Result Grid | Form Editor | Field Types

```
186  
187      -- Total Revenue by Region  
188 •  select Region, sum(Quantity * `Price Per Unit`) as `Total Revenue` from customer_reviews_2  
189      GROUP BY Region;
```

< >

Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

	Region	Total Revenue
▶	North	2385
	East	4400
	West	5955
	South	8020
	NA	5605

Result Grid | Form Editor | Field Types

```
190  
191      -- Total counts of Rating  
192 •  select Rating, count(Rating) as `Total Count` from customer_reviews_2  
193      GROUP BY Rating;
```

< >

Result Grid | Filter Rows: [] | Export: | Wrap Cell Content: |

Rating	Total Count
Good	6
Excellent	6
Poor	6
Average	6
Mischief	1
Worthy	1
Spy	1
Leader	1

Result Grid | Form Editor | Field Types

```
194
195    -- Top Product by Quantity
196 • select Product, sum(Quantity) as `Total Quantity`
197 from customer_reviews_2
198 group by Product order by `Total Quantity` desc limit 1;
```

< >

Result Grid | Filter Rows: Export: Wrap Cell Content: Fetch rows:

Product	Total Quantity
Leadership Manual	85

Result Grid | Form Editor | Field Types

Result 128 × Read Only

199

```
200    -- Top Person by Total Revenue
201 • select Name, sum(Quantity * `Price Per Unit`) as `Total Revenue`
202   from customer_reviews_2
203   group by Name order by `Total Revenue` desc limit 1;
```

< >

Result Grid | Filter Rows: Export: Wrap Cell Content: Fetch rows:

	Name	Total Revenue
▶	Steve Rogers	3365

Result Grid | Form Editor | Field Types

```
205 -- Yearly trend in sale
206 • select year(Date) as Year, sum(Quantity) as `Total Quantity`
207 from customer_reviews_2
208 GROUP BY Year;
```

< []

Result Grid | Filter Rows: [] Export: [] Wrap Cell Content: []

	Year	Total Quantity
▶	2021	275
	2022	450
	2023	240

```
202     from customer_reviews_2
203     group by Name order by `Total Revenue` desc limit 1;
204
205     -- Yearly trend in sale
206 •  select year(Date) as Year, sum(Quantity) as `Total Quantity`
207     from customer_reviews_2
208     GROUP BY Year;
209
210 /*
211 Date: 3 March 2025
212 Name: Sanket Thange
213 */
```

< >

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

	Year	Total Quantity
▶	2021	275
	2022	450
	2023	240



Result Grid



Form Editor



Field Types